基于项目概率分布的协同过滤推荐算法*

王　永[1]　邓江洲[1]　邓永恒[1]　张　璞[2]
[1](重庆邮电大学电子商务与现代物流重点实验室　重庆 400065)
[2](重庆邮电大学计算机学院　重庆 400065)

摘要: 【**目的**】解决传统项目相似性度量方法必须依赖于共同评分项，及传统方法在稀疏数据集中预测准确性不高的问题。【**方法**】将信号处理领域的 KL 散度引入项目相似性的计算中，利用评分值的概率密度分布计算项目相似性，可更有效地发现目标项目的相似邻居项目。【**结果**】在 MovieLens 数据集上的实验结果表明，该算法的推荐综合值 F1 超过 0.65，在预测有效性、预测误差和推荐准确性等方面的评测结果均明显优于当前常用的项目相似性方法。【**局限**】只考虑了项目评分值的比率，未充分利用项目的绝对评分值。【**结论**】算法有效地利用了数据集内的评分信息，较好地克服了数据的稀疏性问题，具有很好的应用价值。

关键词: 项目相似性　协同过滤　KL 散度　推荐算法
分类号: TP391　G350

1　引　言

随着互联网和移动互联网的普及与深度应用，信息量激增。如何解决信息过载，满足用户的个性化需求成为当前的一个研究热点。基于协同过滤的推荐算法是该领域研究中的一种非常有效的方法，得到越来越多的重视。1992 年 Goldberg 等[1]首次提出了协同过滤的概念，目前基于协同过滤的推荐系统已在社交网络和电子商务等领域广泛应用[2-4]。协同过滤推荐算法主要是通过在大量用户群中找到与当前用户相似的用户，以这些相似用户的偏好为依据，为当前用户推荐产品或者服务。

目前，主流的协同过滤推荐算法分为基于用户的协同过滤[5]和基于项目的协同过滤[6-9]两类。在推荐系统中，所采用的用户(项目)相似性度量方法会直接影响推荐质量。传统的基于用户的相似性度量方法[5]，如余弦相似性、皮尔逊相关系数等，虽然取得了巨大的成功，但随着应用环境的变化和深入，稀疏性和冷启动问题日益凸显出来。为了解决这些问题，一些新的相似性方法被提出。Luo 等[10]通过结合两种相似性计算方法解决稀疏数据集问题，提出基于惊异向量的局部用户相似性和全局用户相似性。Ahn[11]提出一个启发式的相似性计算方法 PIP。PIP 方法虽然在某种程度上较好地解决了冷启动问题，但在稀疏的数据集中，由于用户共同评分的项目很少会导致该方法计算的结果不够准确。Bobadilla 等[12]提出的 JMSD 方法是将 Jaccard[13]和 MSD[14]两种方法相结合。该方法弥补了 Jaccard 未考虑绝对评分值和 MSD 忽略了共同评分项目比例的不足。Arwar 等[6]提出一系列基于项目的协同过滤推荐算法，并在实践中取得了较大的成功。然而，当用户间的共同评分项目较少时，上述方法均存在推荐质量不高的问题，即稀疏性问题。为了充分利用每个项目的评分，Patra 等[15]提出基于巴氏系数的相似性度量方法。该方法从概率密度分布的角度计算项目间的相似性，弥补了传统相似性度量方法需要依赖于共同评分项目的不足，对解决稀疏性问题有积极的作用。

本文借鉴文献[15]从概率密度分布角度计算项目

通讯作者：王永，ORCID：0000-0002-5247-043X，E-mail：wangyong_cqupt@163.com。

*本文系国家自然科学基金项目“结合知识图谱的概率话题模型研究”(项目编号:61502066)和重庆市基础与前沿项目“面向产品评论的细粒度观点挖掘方法研究”(项目编号: cstc2015jcyjA40025)的研究成果之一。

间相似性的思路，将信息论中的 KL 散度引入到相似性计算中，提出一种基于项目概率分布的协同过滤推荐算法。基于 KL 散度计算不同项目间的相似性，有效避免了已有方法必须依赖于数据集中共同评分项的不足;利用相似性预测用户对未评分项目的评分；根据预测值，产生推荐数据集。本方法能有效地解决协同过滤算法中普遍存在的数据集稀疏性问题，有很好的实际应用价值。

2 项目相似性度量方法

在推荐系统中，用户评分数据可以表示为如表 1 所示的评分矩阵 $R_{m\times n}$。m 为用户个数，n 为项目个数，r_{ui} 为第 u 个用户对第 i 个项目的评分值。

表 1 用户/项目评分矩阵

User/Item	I_1	I_2	…	I_j	…	I_n
U_1	r_{11}	r_{12}	…	r_{1j}	…	r_{1n}
U_2	r_{21}	r_{22}	…	r_{2j}	…	r_{2n}
…	…	…	…	…	…	…
U_u	r_{u1}	r_{u2}	…	r_{uj}	…	r_{un}
…	…	…	…	…	…	…
U_m	r_{m1}	r_{m2}	…	r_{mj}	…	r_{mn}

相似性计算是推荐算法中最为重要的一个步骤。常用的项目相似性度量方法主要有：余弦相似性、修正余弦相似性、皮尔逊相关系数以及约束皮尔逊相关系数[6]。

(1) 余弦相似性(COS)

在基于项目的协同过滤推荐算法中，项目之间余弦相似性的计算公式如下:

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U}(r_{ui}\times r_{uj})}{\sqrt{\sum_{u\in U}(r_{ui})^2}\sqrt{\sum_{u\in U}(r_{uj})^2}} \tag{1}$$

其中，i, j 为两个不同的项目，U 表示已对项目 i 和 j 共同评分的用户集，u 表示单个用户。

(2) 修正余弦相似性(ACOS)

余弦相似性中没有考虑不同用户之间的评分差异性。在修正余弦相似性中，通过减去项目的平均评分值修正此差异性。对应的计算公式如下:

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U}(r_{ui}-\bar{r}_i)(r_{uj}-\bar{r}_j)}{\sqrt{\sum_{u\in U}(r_{ui}-\bar{r}_i)^2}\sqrt{\sum_{u\in U}(r_{uj}-\bar{r}_j)^2}} \tag{2}$$

其中，$\bar{r}_i$ 表示第 i 个项目的平均评分值。

(3) 皮尔逊相关系数(PCC)

该方法通过减去用户的平均评分值进行相似性结果修正，对应公式如下:

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U}(r_{ui}-\bar{r}_u)(r_{uj}-\bar{r}_u)}{\sqrt{\sum_{u\in U}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{u\in U}(r_{uj}-\bar{r}_u)^2}} \tag{3}$$

$\bar{r}_u$ 表示第 u 个用户对其所评项目的平均值。

(4) 约束皮尔逊相关系数(CPC)

由于皮尔逊相关系数没有考虑用户对项目评分好坏的影响，从而导致用户间评分看似相似(或不同)，但实际的相似度却很低(或很高)。约束皮尔逊相关系数就是为了避免该问题而提出的，其公式为:

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U}(r_{ui}-\bar{r}_{med})(r_{uj}-\bar{r}_{med})}{\sqrt{\sum_{u\in U}(r_{ui}-\bar{r}_{med})^2}\sqrt{\sum_{u\in U}(r_{uj}-\bar{r}_{med})^2}} \tag{4}$$

其中，$\bar{r}_{med}$ 表示评分区间的中值。

上述相似性度量方法虽然广泛应用在推荐算法中，但随着应用环境的变化和深入，其局限性也逐步显现出来，主要表现为：不适宜处理非线性的情况；不能很好地解决数据稀疏性问题。

3 推荐算法

本文方法包括两个主要部分：基于 KL 散度的项目相似性计算；产生推荐。相似性计算是算法最核心的部分，笔者将 KL 散度引入到相似性计算中，有效提高了项目间相似性的适用性和准确性。

3.1 基于 KL 散度的相似性

(1) KL 散度

KL 散度(Kullback-Leibler Divergence)又称 KL 距离，是信息论中统计变量间独立性的重要指标。从概率分布的角度衡量两个变量之间的距离[16-17]。在连续区间 D 中，假设ρ_1和ρ_2分别为两个不同的概率密度函数，则 KL 散度定义为[16]:

$$D(\rho_1\|\rho_2)=\int_D \rho_1(x)\log_2\frac{\rho_1(x)}{\rho_2(x)} \tag{5}$$

对于离散变量，KL 散度定义为:

$$D(\rho_1\|\rho_2)=\sum_{x\in D}\rho_1(x)\log_2\frac{\rho_1(x)}{\rho_2(x)} \tag{6}$$

其中，$\rho_1(x)>0$，$\rho_2(x)>0$，且规定 $0\log_2\sum\frac{0}{\rho}=0$。

KL 散度的优势在于可区别几何距离难以区别的对象。假设图 1 中的对象 1 和对象 2 分别服从正态分布和均匀分布，且两个对象的样本点之间存在大量的重叠。显然，使用几何距离难以区分两个对象。然而，从概率分布角度，使用 KL 散度却能高效地区分它们。

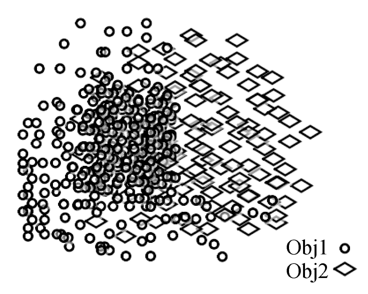


图 1　满足不同概率分布的不同对象示例[16]

(2) 相似性计算

①KL 相似性

在用户评分矩阵中，对任意两个项目 i 和 j，将所有用户对它们的评分视作两个变量序列，可得项目 i 与 j 的 KL 距离 D(i, j)的计算公式如下：

$$D(i,j)=D(\rho_i\parallel\rho_j)=\sum_{v=1}^{r}\rho_{iv}\log_2\frac{\rho_{iv}}{\rho_{jv}} \tag{7}$$

其中，ρ_i 为项目 i 的概率密度函数，r 为评分的最大值，$\rho_{iv}=\frac{\#v}{\#i}$ 为项目 i 中评分值为 v 的比率，#i 是所有用户对项目 i 评分的个数，#v 是所有用户对项目 i 评分值为 v 的个数。

根据 KL 距离，给出基于 KL 的相似性计算公式如下：

$$KL(i,j)=sim(i,j)=\frac{1}{1+D(i,j)} \tag{8}$$

其中，KL 距离越小，项目间相似性越高。

基于 KL 的相似性计算方法不依靠于共同评分项，适用于一些传统相似性方法无法使用的情形。现以一个示例说明本文方法的优势。设项目 i 和 j 的评分分别为：$i=(1,0,2,0,3,0)^T$ 和 $j=(0,3,0,2,0,1)^T$，评分区间为 1-3。由于没有任何用户同时对两个项目同时评分，因此已有的一些方法(如余弦相似性等)无法计算两项目的相似性。然而，根据公式(8)，可以得到项目 i 和 j 的 KL 相似性如下：

$$KL(i,j)=\frac{1}{1+\sum_{v=1}^{3}\rho_{iv}\log_2\frac{\rho_{iv}}{\rho_{jv}}}=\frac{1}{1+(\frac{1}{3}\times\log_2{}^1+\frac{1}{3}\times\log_2{}^1+\frac{1}{3}\times\log_2{}^1)}=1$$

②平滑处理

为了确保 KL 距离能够适用于用户评分矩阵，即保证概率密度函数 $\rho(x)$ 均大于 0，对其进行平滑修正如下：

$$\hat{\rho}(x)=\frac{\rho(x)+\delta}{1+\delta|D|} \tag{9}$$

其中，0<δ<1，$|D|$ 表示离散区域中所有取值的个数。平滑处理后，误差分析如下：

$$|\hat{\rho}(x)-\rho(x)|=\left|\frac{\rho(x)+\delta-\rho(x)-\delta\rho(x)|D|}{1+\delta|D|}\right|=\left|\frac{1-\rho(x)|D|}{1/\delta+|D|}\right|\in\left[\frac{1}{1/\delta+|D|},\frac{|D|-1}{1/\delta+|D|}\right]$$

当 δ 值足够小时，平滑处理后能提供任意精度的相似性估计。

③对称性修正

由公式(7)可知，KL 距离不具有对称性，即 $D(\rho_i\parallel\rho_j)\neq D(\rho_j\parallel\rho_i)$。将其表示两个项目间的距离时需要具有对称性。为此，对 KL 距离进行对称性修正如下：

$$D_S(i,j)=(D(\rho_i\parallel\rho_j)+D(\rho_j\parallel\rho_i))/2 \tag{10}$$

在计算项目间的 KL 相似性时，用 $D_s(i, j)$取代公式(8)中的 D(i, j)。

3.2　产生推荐

(1) 形成最近邻居集。根据公式(8)计算任意项目之间的相似性值，进而得到项目的相似性矩阵$[S_{i,j}]_{n\times n}$，如下所示：

$$S=\begin{bmatrix}S_{1,1}&S_{1,2}&\cdots&S_{1,n}\\S_{2,1}&S_{2,2}&\cdots&S_{2,n}\\\cdots&\cdots&\cdots&\cdots\\S_{n,1}&S_{n,2}&\cdots&S_{n,n}\end{bmatrix}$$

此处，$S_{i,j}(1\leqslant i\leqslant n, 1\leqslant j\leqslant n)$为项目 i 和 j 之间相似性值。

根据项目相似矩阵 S，可得到项目 i 的 N 个最近邻居项目集合 $N_i=\{i_1,i_2,\cdots,i_N\}$，且集合内部元素之间的排序满足 $S_{i,i_j}\geqslant S_{i,i_{j+1}}$。

(2) 预测值计算。通过集合 N_i 中项目的评分，计算目标用户 u 对项目 i 的预测值 $P_{u,i}$，对应的公式如下：

$$P_{u,i}=\frac{\sum_{j\in N_i}S_{i,j}\cdot r_{uj}}{\sum_{j\in N_i}|S_{i,j}|} \tag{11}$$

其中，r_{uj} 为用户 u 对项目 j 的评分。

(3) Top-N 推荐。根据预测值，可以做 Top-N 推荐，即取预测值最高的前 N 个项目作为用户的推荐项目集。

4　算法分析

(1) 稀疏性分析

传统的相似性计算方法，如 ACOS、PCC 和 CPC

等，它们的局限在于“必须依赖于共同评分项”，即至少需要一个用户对项目 i 和 j 同时进行评分。一旦没有共同评分项，传统的方法将无法计算这两个项目间的相似性。这种情况在稀疏的数据集中表现尤为突出。本文提出的方法是利用项目 i 和 j 的评分值的概率分布进行相似性计算，无须依赖共同评分项，而且对用户评价项目的数量也没有要求。因此，即使是在稀疏的数据集中，采用本文方法也能获取必要的信息，完成项目相似性的计算。所以，本文提出的方法能更好地应对推荐算法中常存在的数据稀疏性问题。

(2) 适用性分析

本文提出的相似性方法是以评分值的概率密度为基础，通过 KL 散度计算项目间的相似性。该方法对数据集中数据的分布没有做任何假设。然而，一些传统的相似性计算方法通常假设两变量间存在线性关系，其适用范围存在局限性。就用户评分数据集来说，其中的数据是离散的，数据之间往往不存在线性关系。若基于线性假设进行预测，必然难以获得好的结果。本文方法对数据间是否存在线性关系没有任何要求，因此具有更好的适应性，既适合处理线性数据关系的问题，也适合处理非线性数据关系的问题。

(3) 信息利用率分析

本文方法不受共同评分项的限制，在计算评分项的概率密度时，会使用评分矩阵中所有的用户评价信息。因此，本文算法对评价信息的利用率高于其他相似性计算方法。高的信息利用率可以避免预测结果的片面性，防止预测结果出现大的波动，从而提高了本文算法的整体性能。

5 实验结果与分析

5.1 数据集

采用公开的数据集 MovieLens①作为本文算法测试和验证的数据集，包括 706 个用户对 8 570 部电影的评分，共有评论记录 100 023 条。从该数据集中选取了 59 775 条评分作为实验数据集，包含 706 个用户和 813 部电影，评分范围为 1-5，且每部电影被用户评分至少 25 次。实验数据集的稀疏度为 10.4%。为了测试推荐算法的性能，将数据集划分为 80%的训练集和 20%的测试集。

5.2 评价指标

推荐算法的评价主要包括预测准确性、推荐准确性和计算有效性三个方面[6]。常用的预测准确性指标为平均绝对误差(MAE)和根均方误差(RMSE)，公式如下：

$$\mathrm{MAE}=\frac{\sum_{i=1}^{n}\left|r_{ui}-\hat{r}_{ui}\right|}{n} \tag{12}$$

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{i=1}^{n}(r_{ui}-\hat{r}_{ui})^2}{n}} \tag{13}$$

其中，r_{ui} 和 $\hat{r}_{ui}$ 分别是用户 u 对项目 i 的实际评分值和预测评分值；n 代表待预测项目的个数。这两项指标的值越小表示预测的准确性越高。

常用的推荐准确性度量指标为：准确率(Precision)、召回率(Recall)和 F1 值，对应的计算公式如下：

$$\mathrm{Precision}=\frac{n(I_p\cap I_a)}{n(I_p)} \tag{14}$$

$$\mathrm{Recall}=\frac{n(I_p\cap I_a)}{n(I_a)} \tag{15}$$

$$\mathrm{F1}=\frac{2\times \mathrm{Precision}\times \mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{16}$$

其中，I_p 为预测推荐的项目个数，I_a 为真实的推荐项目个数。F1 值是综合了准确率和召回率的评价指标，其值越大，说明推荐的综合性能越好。在本文实验中，以大于用户平均评分值作为项目推荐的标准，据此确定推荐的项目列表。

此外，计算有效性的评价指标为：有效预测数和完美预测数。有效预测数是指根据预测值的计算公式，能够从用户评分数据集中成功算出预测值的总数量。完美预测数是指计算出的预测值与真实评分值相同的总数量。

5.3 结果分析

为了与本文算法进行对比，对 ACOS、PCC 和 CPC 等方法进行对比测试。同时，由于不同的邻居个数 K 对测试结果有不同的影响，因此，在本文的实验中也对其进行考虑。

①http://www.grouplens.org.

(1) 有效预测数和完美预测数分析

在实验数据集上，预测的项目总数为 12 017 个。由图 2 可知，不管计算过程中选择的邻居个数 K 如何变化，本文算法的有效预测数和完美预测数均最高。这说明本文算法比 ACOS、PCC 和 CPC 等方法的适应性更好，可以在更多的数据条件下计算出有效的预测值且准确性更高。

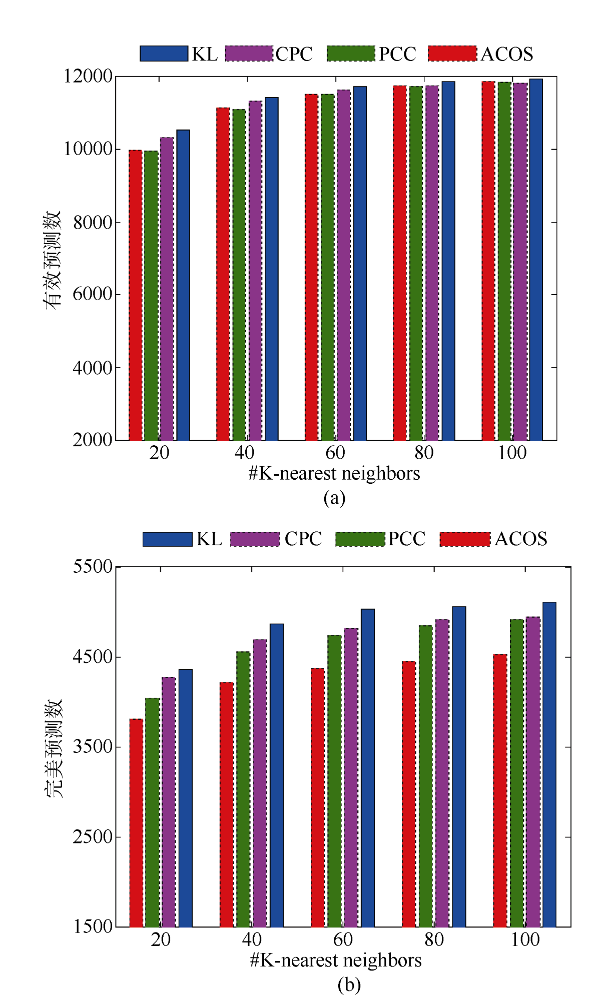


图 2　有效预测数与完美预测数的比较结果

(2) MAE 和 RMSE

MAE 和 RMSE 主要反映的是预测评分值与实际评分值之间的偏差。在图 3 中，可以看出本文算法的 MAE 和 RMSE 优于各传统的相似性方法，两种误差值在整体上都比其他相似性方法更低。随着 K 值的增加，MAE 和 RMSE 均缓慢减少，总体的范围为：0.739≤MAE≤0.779，0.974≤RMSE≤1.049，这表明本文算法的推荐精度较好。

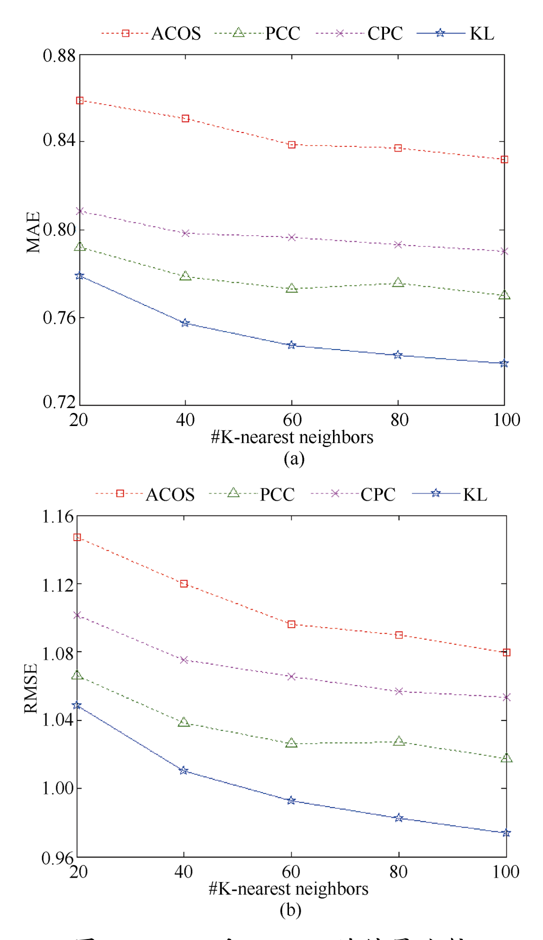
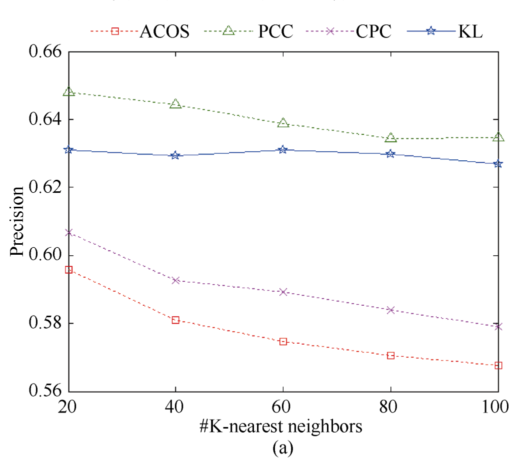


图 3　MAE 和 RMSE 的结果比较

(3) Precision、Recall 和 F1

在图 4(a)中，PCC 方法的准确率最高，其次为本文方法，且两者之间相差不大。在图 4(b)中，无论 K 值如何变化，KL 相似性方法的召回率均明显优于其他方法。F1 值是综合考虑准确率和召回率的指标。从图 4(c)可知，本文算法的 F1 值明显优于其他方法。综上分析可得本文算法具有更好的推荐性能。



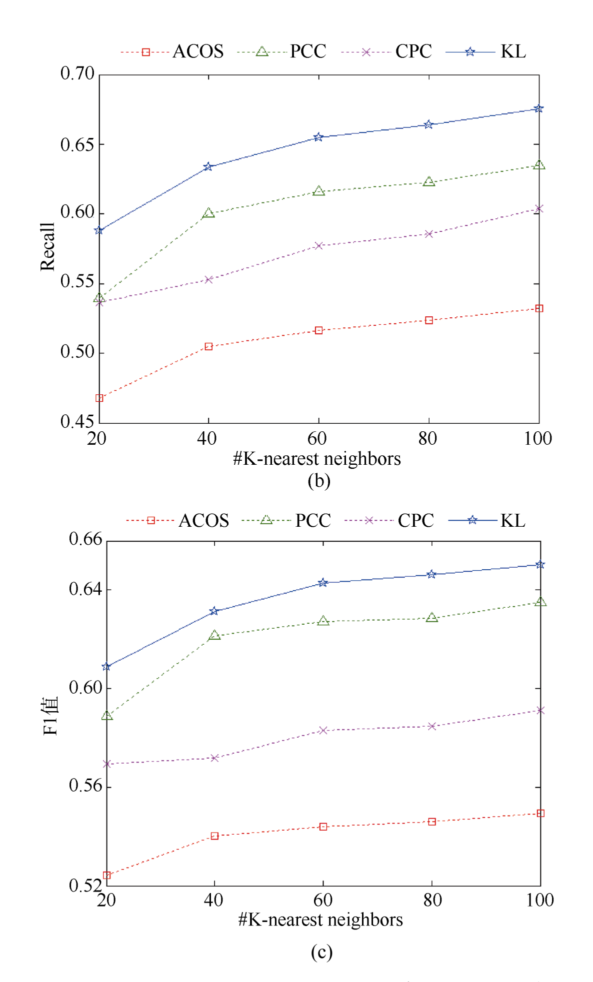


图 4　Precision、Recall 和 F1 值的结果比较

6　结　语

本文将信息论中的 KL 散度引入到协同过滤算法的相似性计算中，提出基于 KL 相似性的协同过滤推荐算法。该方法利用评分值的概率密度分布计算项目之间的相似性。其优势在于它对用户的项目评价数量没有要求，也不要求用户同时对多个项目进行评分。限制条件的放宽，意味着本文方法能找到更多满足其计算条件的评分数据，即便是在稀疏数据集中也能有效完成预测值的计算和项目推荐。因此，与传统的相似性计算方法相比，本文方法更好地解决了数据稀疏性问题。在 MovieLens 公开数据集中的实验表明，本文基于 KL 相似性的协同过滤算法优于其他类似方法，有效提高了整体的推荐质量。

参考文献:

[1] Goldberg D, Nichols D, Oki B M, et al. Using Collaborative Filtering to Weave an Information Tapestry [J]. Communications of the ACM, 1992, 35(12): 61-70.

[2] Zheng N, Li Q, Liao S, et al. Which Photo Groups Should I Choose a Comparative Study of Recommendation Algorithms in Flickr [J]. Journal of Information Science, 2010, 36(6): 733-750.

[3] Brynjolfsson E, Hu Y J, Smith M D. Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers [J]. Management Science, 2003, 49(11): 1580-1596.

[4] Breese J, Hecherman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering [C]. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998.

[5] Xu C, Xu J, Du X. Recommendation Algorithm Combining the User-based Classified Regression and the Item-based Filtering [C]. In: Proceedings of the International Conference on Electronic Commerce: The New E-commerce-Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet, Fredericton, New Brunswick, Canada. 2006: 574-578.

[6] Arwar B, Karypls G, Konstan J, et al. Item-based Collaborative Filtering Recommendation Algorithms [C]. In: Proceedings of the 10th International World Wide Web Conference. 2001.

[7] Kim B M, Li Q, Park C S, et al. A New Approach for Combining Content-based and Collaborative Filters [J]. Journal of Intelligent Information System, 2006, 27(1): 79-91.

[8] Karypis G. Evaluation of Item-based Top-N Recommendation Algorithms[C]. In: Proceedings of the 10th International Conference on Information and Knowledge Management. 2001.

[9] Deng A, Zhu Y, Shi B. A Collaborative Filtering Recommendation Algorithm Based on Item Rating Prediction [J]. Journal of Software, 2003, 14(9): 1621-1628.

[10] Luo H, Niu C, Shen R, et al. A Collaborative Filtering Framework Based on both Local User Similarity and Global User Similarity [J]. Machine Learning, 2008,72(3): 231-245.

[11] Ahn H J. A New Similarity Measure for Collaborative Filtering to Alleviate the New User Cold-Starting Problem [J]. Information Sciences, 2008, 178 (1): 37-51.

[12] Bobadilla J, Ortega F, Hernando A, et al. A Collaborative Filtering Approach to Mitigate the New User Cold Start Problem [J]. Knowledge-Based Systems, 2012, 26: 225-238.

[13] Koutrica G, Bercovitz B, Garcia H. FlexRecs: Expressing and Combining Flexible Recommendations [C]. In: Proceedings

of the ACM SIGMOD International Conference on Management of Data. 2009.
[14] Cacheda F, Carneiro V, Fernández D, et al. Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender System [J]. ACM Transactions on the Web, 2011, 5(1): 1-33.
[15] Patra B K, Launonen R, Ollikainen V, et al. Exploiting Bhattacharyya Similarity Measure to Diminish User Cold-start Problem in Sparse Data [A]. // Discovery Science [M]. Springer International Publishing, 2014: 252-263.
[16] Kullback S, Leibler R A. On Information and Sufficiency [J]. The Annals of Mathematical Statistics, 1951, 22(1): 79-86.
[17] Huang A. Similarity Measures for Text Document Clustering [C]. In: Proceedings of the 6th New Zealand Computer Science Research Student Conference. 2008.

作者贡献声明:

王永: 确定研究目标和技术路线, 提出论文修订意见, 修改论文;
邓江洲: 算法设计, 起草、修改论文;
邓永恒: 数据收集, 实验分析;
张璞: 辅助算法设计, 算法性能提升。

利益冲突声明:

所有作者声明不存在利益冲突关系。

支撑数据:

支撑数据见期刊网络版 http://www.infotech.ac.cn。
[1] 王永, 邓江洲. ACOSsim.xlsx. 修正余弦相似性计算结果.
[2] 王永, 邓江洲. PCCsim.xlsx. 皮尔逊相关系数计算结果.
[3] 王永, 邓江洲. CPCsim.xlsx. 约束皮尔逊相关系数计算结果.
[4] 王永, 邓江洲. KLsim.xlsx. 基于 KL 散度的项目相似性计算结果.
[5] 王永, 邓江洲. MAE_RMSE.xlsx. 相关误差计算结果.
[6] 王永, 邓江洲. F1.xlsx. F1 值计算结果.
[7] 王永, 邓江洲. 有效预测数和完美预测数.xlsx. 有效和完美预测数计算结果.

收稿日期: 2016-01-26
收修改稿日期: 2016-03-23

A Collaborative Filtering Recommendation Algorithm Based on Item Probability Distribution

Wang Yong[1]　Deng Jiangzhou[1]　Deng Yongheng[1]　Zhang Pu[2]
[1](Key Laboratory of Electronic Commerce and Logistics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)
[2](College of Computer Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: [**Objective**] This study tries to reduce the reliance of co-rated items in the traditional item similarity measurements and then improve the prediction precision of the sparse datasets. [**Methods**] First, we modified the Kullback-Leibler (KL) divergence from the signal processing domain to compute item similarities. Second, we calculated the similarity with the help of density distribution of ratings, and then found the neighboring items more effectively. [**Results**] We examined the proposed algorithm on MovieLens and the achieved F1 measure value was over 0.65. The accuracy, efficiency and error rates of the new prediction mechanism were much better than traditional item similarity measurements. [**Limitations**] The proposed algorithm considered the density of ratings, however, it did not utilize the absolute value of item ratings. [**Conclusions**] The proposed algorithm effectively uses the rating information to address the sparse dataset issue. Thus, it has strong potentiality in practice.

Keywords: Item similarity　Collaborative filtering　Kullback-Leibler divergence　Recommendation algorithm